

REMARKS

This responds to the Office Action mailed on December 2, 2008.

Claims 47-66 are canceled, and claims 67-78 are added; as a result, claims 67-78 are now pending in this application.

Interview Summary

Applicant thanks Examiner Insun Kang for the courtesy of a personal interview on January 8, 2009 with Applicant's representatives John I. Fischer and Steven W. Lundberg where the proposed claims were discussed in view of Van Hoff. No agreement was reached with respect to the proposed claims.

§ 102 Rejection of the Claims

Claims 47-49, 53-61 and 63-66 were rejected under 35 U.S.C. § 102(e) as being anticipated by Van Hoff et al. (U.S. 6,272,536; hereinafter "Van Hoff"). Applicant has canceled claims 47-66, thereby obviating the rejections of these claims.

§ 103 Rejection of the Claims

Claims 50-52 and 62 were rejected under 35 U.S.C. § 103(a) as being obvious over Van Hoff. Applicant has canceled claims 47-66, thereby obviating the rejections of these claims.

New Claims 67-78

Claims 67-78 have been added in this response. Applicant respectfully submits that these claims are fully supported by the specification as originally filed, and that no new matter has been added. These claims recite a system and method for distributing tasks from a distribution server to a client computer.

Applicant's application describes and Applicant has claimed a task distribution system wherein at least part of the infrastructure to distribute tasks is installed as part of the claimed invention. Tasks are broadly described as including but not limited to being a file, a script, or a command that is run or installed on a target computer. Tasks of these types are assigned to

computers, users, or groups of computers or users via an administrative user interface. The user interface may be presented with functional, familiar, and easy-to-use controls, such as collapsible/expandable hierarchical lists.

Tasks are highly-configurable and may be imbued with rules, configuration settings, and prerequisites. For example, for a task to execute at a computer there may be a prerequisite condition, such as a particular value in a registry entry. Other rules may be associated with the task, such as when to execute or how many times a particular task is allowed to execute on a target computer.

Once a task is available, it can be assigned to one or more computers, or one or more users. This configurability is highly advantageous. For example, when an administrator assigns a task to a group of computers, whenever a computer is added to the group, it automatically is pushed the tasks assigned to that group. Conversely, when a computer leaves a group, the computer will sever the associations with the tasks assigned to the group. As another example, when an administrator assigns a task to a particular user, no matter which computer the user is at, the user will receive the assigned task. Such user-specific tasks may be executed as the user logs on or logs off, as appropriate in view of the assigned task.

In addition, tasks may be executed regardless of the current user's permissions. This type of execution provides additional layers of security for system administrators and others. For example, a user may be restricted from accessing or modifying a computer's registry. However, when a task needs to execute with a special user account, the task may be configured to log on as the special user and execute the task. Other mechanisms may be used, such as a "run as" function. Tasks may also be scheduled to run at a particular time or with a recurrence.

As indicated above, the invention as claimed includes installing at least a portion of an infrastructure used to distribute tasks. This installation can be performed remotely. The remote installation can also optionally be performed without local user intervention. Such remote installation may be performed either directly or indirectly. A direct mechanism may include using an administrative share (e.g., C\$) and registry entries to provide support for a direct installation. An indirect mechanism may include the use of a "subscription file" that is delivered to potential client. When a user accesses a subscription file, the user is provided the opportunity to install the client. The availability of several approaches to rolling out the client on first usage

removes a serious impediment to installation of software, particularly in a network environment, and allows much greater efficiency and consistency and speed from start to finish of an installation process.

Although remote assignment of tasks and remote installation of the client software provide certain security measures, various embodiments of the claimed invention also include additional security measures. One additional security measure is the use of licenses. Licenses may be used to restrict client operability or server access. When a client communicates with a distribution server, a license is checked to ensure that no two computers on the network are using the same license. In addition, licenses may be distributed from a central authority, thereby increasing the security of the distribution mechanism.

When compared to Van Hoff, Applicant respectfully submits that Van Hoff does not disclose or describe elements found in these claims. Van Hoff is related to a system and method for distributing software applications and data to clients over a network.¹ The system includes several distinct parts, including a client called a “tuner,” a server called a “transmitter” and applications that are called “channels.”² The end-user uses the tuner to subscribe to channels.³ A request is created and transmitted to the server, and in response, the server transmits the code and data for the channel to the client.⁴ Periodically, the tuner can connect to the server to obtain updates for the channel.⁵ With the tuner, an end-user can subscribe to a number of channels, where channels may not be serviced by the same server.

Importantly, as best as Applicants can determine, there is *no disclosure* in Van Hoff of how the tuner process is first installed on a client.

Several differences between Van Hoff and Applicant’s innovation are immediately apparent. In particular, where Van Hoff relies on the user to subscribe to a channel (application) and request the associated code and data, Applicant discloses and claims a system and method where tasks are assigned to client computers from an administrative user. In addition, where Van Hoff is silent as to the mechanism of installing the tuner, Applicant discloses and claims a remote installation procedure. In addition, where Van Hoff is silent to the organization,

¹ Van Hoff at Abstract.

² *Id.*

³ *Id.* at col. 3, line 14.

⁴ *Id.* at col. 3, lines 14-17.

⁵ *Id.* at col. 3, lines 22-25.

presentation, and implementation of an administrative interface to manage its system, Applicant discloses and claims a user-friendly, high-functional hierarchical interface for assigning and managing task associations. Other advantages and differences become apparent in the detailed claim analysis, as follows.

Accordingly, as discussed further below, the Applicant's invention distinguishes over the teachings of Van Hoff in at least three important areas:

- 1) the Applicant's invention as claimed includes methods and systems for installing at least a portion of the task distribution infrastructure;
- 2) the Applicant's invention as claimed assigns tasks from an administrative authority as opposed to the clients selectively choosing to subscribe to channels; and
- 3) the Applicant's provide an advantageous mechanism to assign tasks, and in particular a hierarchical interface, as described in the claims.

Independent claim 67:

Independent claim 67 recites "A system comprising one or more distribution servers used to install client software on a plurality of client computers and to subsequently use the client software to perform tasks later assigned to the one or more of the client computers by at least one of the distribution servers." By installing client software using the distribution servers, the system described in claim 67 bypasses the operational complexity and expense and time of distributing the client software by manual means. The system as claimed also provides a centralized mechanism for distributing tasks, installing client software, and controlling related operations.

Claim 67 continues with "at least one computer program operative on the one or more distribution servers to communicate with a plurality of client computers to initiate an installation of client software on the client computers."

As further recited in claim 69, the "installation on one of the plurality of client computer comprises" operations, including "distributing a client software component installation file to the client computer." Another operation for installation includes "distributing a client service installation file to the client computer." The client software component acts in conjunction with the client service, both of which are installed using separate files. The service installation file

may be used to install the service and the client software component on the target client machine without a user logged onto the target client machine. In another embodiment, the service installation file may be used to install the client software component and the service without regard to the user's permissions on the client machine. These features are illustrated, at least in part, in dependent claim 69, which is discussed *infra*.

Another operation for installation includes "from at least one of the distribution servers, remotely causing the client software component installation file and the client service installation file to be used to install a client software component and a client service, respectively." Again, remote installation is useful and advantageous for at least the reason of providing a centralized point of control and management. In addition, by remotely installing the client software and the service, and bypassing the user's permissions, the system can provide high functionality with minimal inconvenience to the users.

As recited in claim 67, "the client software component and the client service are associated with one another when installed on the client computer." For example, the client software component may provide a user interface whereas the client service operates at the operating system level to provide lower level functionality, such as communication, synchronization, and control of task distribution operations.

Further, as recited in claim 67, "the client software component and the client service are subsequently used by the client computer to perform at least in part one or more tasks assigned to the client computer by at least one of the distribution servers." By controlling the distribution of the client software component and the client service, the administrator is provided with an additional security measure. In particular, if the client software component and the client service were generally available, unauthorized computers may be loaded with the client software and provide an attack vector against the distribution servers.

Claim 67 continues with "the one or more distribution servers including one or more distribution computer programs are operative to" perform operations, including "manage a hierarchical list of at least some of the plurality of client computers, wherein the hierarchical list uniquely identifies each of the client computers on the hierarchical list using a computer identification and wherein the hierarchical list is configured to arrange and group the at least some of the plurality of client computers into one or more computer groups." Moreover, as

recited in claim 67, the distribution computer programs are further operative to “assign tasks to one or more client computers managed in the hierarchical list, wherein the tasks are assigned to individual ones of the client computers on the hierarchical list based on the computer identification or to groups of computers on the hierarchical list based on the computer groups, wherein the tasks include at least one item selected from the group: a file, a script or a command.” In addition, as recited in claim 67, the distribution computer programs are further operative to “store the tasks assigned to the client computers in the hierarchical list.”

By using a hierarchical list, tasks and target objects may be divided into two separate root nodes to avoid ambiguity between existence and assignment. For instance, one or more tasks may be organized under a group of tasks, which exist under a “task” root node. Providing groupings of tasks can simplify assignment to target computers, such as with drag-and-drop of the group node to one or more target computers. Similarly, one or more computers may be grouped using a “computer group” node, which exists under a “computer” root node. In this way, dragging one or more tasks, or task groups, to a computer group node would assign the one or more tasks to all computers associated with the computer group node. The ease of administration and the visual organization and simplicity illustrates at least some of the advantages of the use of hierarchical lists.

In contrast, Van Hoff is silent as to how the client application (e.g., the “Tuner” as described in Van Hoff at col. 3, lines 5-8; or “tuner process 152” as described in Van Hoff at col. 4, line 57 to col. 5, line 9) is installed and clearly Van Hoff is silent with regard to remote installation of such a client application.

Moreover, the tuner process 152 described in Van Hoff is exclusively a “pull” mechanism operated by the user of the client machine. As described in Van Hoff:

Client system 140 uses its operating system 150 to run a tuner process 152 to subscribe to one or more channels from a transmitter. Tuner process 152 can be invoked directly by the end user, or via a web browser 151. When the tuner process 152 is first invoked by the end user it may select an anonymous identifier which can be used to identify this instance of the tuner process in subsequent communications. In the preferred embodiment a 128 bit random number is used as the identifier.

To obtain the initial channel data, tuner process 152 uses network interface 141 to send a subscribe request to transmitter process 121 running on server system 110. Transmitter process 121 will respond with an update reply containing channel data 133 as will be described in more detail below. In response, tuner process 152 stores channel data 161 in storage system 154. Once the channel data is loaded tuner process 152 can start channel application 153. Channel application 153 is the active application corresponding to a channel 159 which is executed by operating system 150 from the code stored in channel data 161.

Van Hoff at col. 4, line 57 to col. 5, line 9. Clearly, Van Hoff refers to a pull environment, whereas Applicant's tasks are assigned to client machines from a distribution server; and thus can be operated in a "push" paradigm. In particular, Van Hoff appears to refer to a user actively subscribing to a channel that corresponds to a channel application, whereas Applicant's claim 67 recite "manag[ing] a hierarchical list of at least some of the plurality of client computers" at a distribution server, and "assign[ing] tasks to one or more client computers managed in the hierarchical list" from the distribution server.

For at least these reasons, Applicant respectfully submits that Van Hoff cannot be used to anticipate these claims and submits that these claims are allowable over Van Hoff.

Dependent claim 68:

Claim 68 depends from claim 67 and recites "the client software component and client service operative on the client computer to" perform operations, including "connect to at least one of the distribution servers."

In addition, the client software component and client service "request one or more scheduled tasks from the at least one distribution servers, the one or more scheduled tasks having been previously assigned to the client computer using the hierarchical list." The tasks assigned to client computers may be scheduled, for example, to push out a software update during a low-usage period to avoid inconveniencing users. Schedules may be configurable to the day, time, and number of occurrences. Such scheduled tasks may avoid network congestion, reduce user inconvenience, and improve task administration.

In addition, the client software component and client service "receive the one or more scheduled tasks, the one or more scheduled tasks including at least one of a file, a script, or a

command.” In addition, the client software component and client service “process the one or more scheduled tasks.” In claim 68, Applicant recites three modalities of delivering tasks: a file, a script, and a command. These different modes of delivering tasks provides flexibility and functionality. For example, a file may be used to install a software program; a script may be used to emulate user interface action; or a command may be used to alter a system setting.

Dependent claim 69:

Claim 69 depends from claim 68 and recites “to process the one or more scheduled tasks, the client software component and client service are operative on the client computer to” perform operations including “determine whether the client computer is licensed to execute the tasks.” Licensing provides an additional layer of security for distribution and administration.

In addition, the client software component and client service “determine whether a special account is required to execute the tasks.” For example, a special account may be an administrator account.

In addition, the client software component and client service “automatically log into the client computer using the special account when the client computer is licensed to execute the tasks and requires the special account to execute the tasks.” In addition, the client software component and client service “use the special account to perform the tasks.” By automatically using the special account, the regular user of the client computer is kept from the installation process. The resulting effect is that system administrators are provided additional assurance that client machines are kept under control.

Independent claim 70:

Independent claim 70 recites a “method for distributing tasks from a distribution server to a client computer.” Claim 70 includes the operation “distributing a client software component installation file to the client computer from the distribution server.”

In addition, claim 70 recites “distributing a client service installation file to the client computer from the distribution server.”

In addition, claim 70 recites “remotely invoking from the distribution server, the client software component installation file and the client service installation file to install a client

software component and a client service, respectively.” As with claim 68, remote installation is advantageous to provide high functionality with minimal inconvenience to the users.

As recited in claim 70, “the client software component and the client service are associated with one another when installed on the client computer”.

In addition, “the client software component and the client service are first installed by the installation on the client computer.”

Claim 70 further comprises “managing a hierarchical list of at least some of a plurality of client computers, wherein the hierarchical list uniquely identifies each of the client computers on the hierarchical list using a computer identification and wherein the hierarchical list is configured to arrange and group the at least some of the plurality of client computers into one or more computer groups and wherein the client computer is included in the plurality of client computers.”

In addition, claim 70 recites “assigning tasks to one or more client computers of the at least some of the plurality of client computers managed in the hierarchical list, wherein the tasks are assigned to individual ones of the client computers on the hierarchical list based on the computer identification or to groups of computers on the hierarchical list based on the computer groups, wherein the tasks include at least one item selected from the group: a file, a script or a command.”

In addition, claim 70 recites “storing the tasks assigned to the client computers in the hierarchical list.”

As recited in claim 70, “at least one of the tasks assigned to one of the client computers is performed at least in part using one or more of the client software component and the client service component installed on the client computers.”

As with claim 68, Applicant respectfully submits that Van Hoff does not disclose how the client software is installed. Moreover, Van Hoff refers to a “pull” environment, whereas Applicant’s claim 70 recites a “push” environment.

Dependent claim 71:

Claim 71 depends from claim 70 and further comprises “connecting to the distribution server via the client software component and the client service.”

In addition, claim 71 recites “requesting one or more scheduled tasks from the distribution server, the one or more scheduled tasks having been previously assigned to the client computer using the hierarchical list.”

In addition, claim 71 recites “receiving the one or more scheduled tasks, the one or more scheduled tasks including at least one of a file, a script, or a command.”

In addition, claim 71 recites “processing the one or more scheduled tasks.”

Claim 71 is similar to claim 68 and Applicant refers the examiner to the discussion *supra*.

Dependent claim 72:

Claim 72 depends from claim 71 and further comprises “determining whether the client computer is licensed to execute the tasks.”

In addition, claim 72 recites “determining whether a special account is required to execute the tasks.”

In addition, claim 72 recites “automatically logging into the client computer using the special account when the client computer is licensed to execute the tasks and requires the special account to execute the tasks.”

In addition, claim 72 recites “using the special account to perform the tasks.”

Claim 72 is similar to claim 69 and Applicant refers the examiner to the discussion *supra*.

Independent claim 73:

Independent claim 73 recites a “system comprising a client computer; and a distribution server communicatively coupled to the client computer, wherein the distribution server is configured to” perform operations.

Claim 73 includes the operation “initiate an installation procedure.”

The “installation procedure” comprises an operation to “distribute a client software installation file to the client computer.”

Further, the “installation procedure” comprises an operation to “remotely invoke the client software installation file to produce a client software installation.”

The client computer is configured such that “upon a successful client software installation, is configured to” perform operations, including “use the client software installation

to” perform operations including “query the distribution server to obtain one or more tasks assigned to the client computer.” As discussed above, Van Hoff appears to refer to a “pull” mechanism where a user “subscribes” to channels. As such, Applicant respectfully submits that Van Hoff does not disclose “tasks assigned to the client computer” much less querying for such tasks, as recited in claim 73.

In addition, the client software installation is configured to “process the one or more tasks assigned to the client computer.”

As with claim 68, Applicant respectfully submits that Van Hoff does not disclose how the client software is installed. Moreover, Van Hoff refers to a “pull” environment, whereas Applicant’s claim 73 recites a “push” environment.

Dependent claim 74:

Claim 74 depends from claim 73. Claim 74 recites that “for a particular task of the one or more tasks” operations of “determine whether the particular task is enabled for a particular user of the client computer” and “execute the task when the task is enabled for the particular user of the client computer” are performed.

In Applicant’s system, tasks can be enabled at the computer level or at an individual user level. By providing additional levels of granularity in task assignment, task management and administration becomes highly configurable. In addition, by limiting the number of users who are provided software, configuration settings, or other task payloads, system administrators may be able to make efficient use of licenses, computer storage, and other network or computer resources. Moreover, limiting the number of changes to a particular user’s operating environment may avoid confusing unsophisticated users and result in fewer support costs.

Dependent claim 75:

Claim 75 depends from claim 73. Claim 75 recites that “for a particular task of the one or more tasks” operations of “determine whether the client computer is licensed to execute the task” and “execute the task when the client computer is licensed to execute the task” are performed.

Claim 75 is similar to claim 69 and Applicant refers the examiner to the discussion *supra*.

Independent claim 76:

Independent claim 73 recites a “method comprising at a distribution server, initiating an installation procedure.”

As recited in claim 76, the “installation procedure” comprises “distributing a client software installation file to a client computer, the client computer coupled to the distribution server.”

Furthermore, the “installation procedure” comprises “remotely invoking the client software installation file to produce a client software installation.”

Furthermore, claim 76 recites “at the client computer, upon a successful client software installation, using the client software installation by: querying the distribution server to obtain one or more tasks assigned to the client computer.”

Furthermore, “upon a successful client software installation,” claim 76 recites “processing the one or more tasks assigned to the client computer.”

As with claim 68, Applicant respectfully submits that Van Hoff does not disclose how the client software is installed. Moreover, Van Hoff refers to a “pull” environment, whereas Applicant’s claim 76 recites a “push” environment.

Dependent claim 77:

Claim 77 depends from claim 76 and recites further details of “processing the one or more tasks assigned to the client computer.” In particular, “for a particular task of the one or more tasks,” “determining whether the particular task is enabled for a particular user of the client computer.”

In addition, claim 77 recites “executing the task when the task is enabled for the particular user of the client computer.”

Claim 77 is similar to claim 74 and Applicant refers the examiner to the discussion *supra*.

Dependent claim 78:

Claim 78 depends from claim 76 and recites further details of “processing the one or more tasks assigned to the client computer.” In particular, “for a particular task of the one or more tasks,” “determining whether the client computer is licensed to execute the task.”

In addition, claim 78 recites “executing the task when the client computer is licensed to execute the task.”

Claim 78 is similar to claim 69 and Applicant refers the examiner to the discussion *supra*.

CONCLUSION

Accordingly, as discussed above, the Applicant’s invention distinguishes over the teachings of Van Hoff in at least three important areas as reiterated below and therefore the claimed invention is neither anticipated nor rendered obvious under Section 103 in view of Van Hoff. In particular:

1) Applicant’s invention as claimed includes methods and systems for installing at least a portion of the task distribution infrastructure, and no such mechanisms are described in Van Hoff – rather Van Hoff only describes the use of infrastructure once installed.

2) Applicant’s invention as claimed assigns tasks from an administrative authority as opposed to the clients selectively choosing to subscribe to channels, as opposed to Van Hoff which teaches a “pull” mechanism paradigm.

3) Applicant provides an advantageous mechanism to assign tasks, and in particular a hierarchical interface, as described in the claims, as opposed to Van Hoff which describes no such mechanisms at all.

Applicant notes that the hierarchical interface was previously presented in claim 65. Although not specifically addressed in the Office Action of December 2, 2008, Applicant submits that the rejection is deficient because Van Hoff does not teach such an interface, as discussed above.

Applicant respectfully submits that the claims are in condition for allowance, and notification to that effect is earnestly requested. The Examiner is invited to telephone Applicant's representative at (612) 371-2134 to facilitate prosecution of this application.

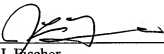
If necessary, please charge any additional fees or deficiencies, or credit any overpayments to Deposit Account No. 19-0743.

Respectfully submitted,

SCHWEGMAN, LUNDBERG & WOESSNER, P.A.
P.O. Box 2938
Minneapolis, MN 55402
(612) 371-2134

Date May 4, 2009

By



John I. Fischer
Reg. No. 60,900

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being filed using the USPTO's electronic filing system EFS-Web, and is addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on this 4th day of May, 2009.

Kathryn McCook

Name

Kathryn McCook

Signature